Hybridization Between Iterative Simulated Annealing and Modified Great Deluge for Medical Clustering Problems

Anmar Abuhamdah
Department of Computer Science
Jearsh private university
Amman, Jordan

Bassam M. El-Zaghmouri
Department of Computer Information Systems
Jearsh private university
Amman, Jordan

Anas Quteishat
Department of Computer Engineering,
Al-Balqa' Applied University
Amman, Jordan

Rawnaq Kittaneh
Department of Computer Science
University Kabangsaan Malaysia
Bangi, Malaysia

Abstract—Clustering is a type of classification under optimization problems, which is considered as a critical area of data mining. Medical clustering problem is a type of unsupervised learning in data mining. This work presents a hybridization between our previous proposed Iterative Simulated Annealing (ISA) and Modified Great Deluge (MGD) algorithms for medical clustering problems. The aim of this work is to produce an effective algorithm for partitioning N objects into K clusters. The idea of the hybridization between MGD and ISA is to incorporate the strength of one approach with the strength of the other, hoping a more promising algorithm. Also, this combination can help to diversify the search space. Experimental results obtained two ways of calculating the minimal distance that have been tested on six benchmark medical datasets show that, ISA-MGD is able to outperform some instances of MGD and ISA algorithms.

Keywords: Clustering; Modified Great Deluge component; Iterative Simulated Annealing.

I. INTRODUCTION

Clustering problem is considered as an important type of unsupervised data mining technique, which divides the input space into K regions based on some similarity/dissimilarity measures, where the K value might not be known a priori [1], [2]. A cluster is a set of data (objects) which are related to each other within the same cluster and unrelated to the data in other clusters [1], [2]. The process of grouping a set of physical data into classes of related data is called clustering [1], [2]. Clustering problem is an NP-hard problem when there are more than three clusters [3]. Recently, the problem with two clusters also considered as an NP-hard problem [4], [5].

Clustering in a problem aims to minimize within the group variance and maximize between the group variance [6]. Recently, there are various algorithms has been applied on numerous domains to solve clustering problems, like Genetic Algorithm [7], [8] and [9], Tabu Search [10], and Artificial Bee Colony [11], iterative simulated annealing [12] and modified great deluge (MGD) [13].

In this work, we use a Multi K- Means algorithm as in [12] to construct the initial solution since it simple and able to deal with a huge number of data patterns.

In this work, we hybridize Modified Great Deluge (MGD) [13] and Iterative Simulated Annealing (ISA) [12] algorithms to diversify the search space of MGD and ISA algorithms and incorporate the strength of each other. The idea of the hybridization (ISA-MGD) is to overcome some of the limitation of MGD and ISA algorithms.

In order to evaluate the performance of the propose algorithm, ISA-MGD applied over six benchmark datasets using two ways of calculation the minimal distance (i.e., between objects and between centers). After that, it’s compared to MGD and extended ISA algorithms that applied by using the same two ways of calculation the minimal distance and the six benchmark datasets [12], [13].

II. PROBLEM DESCRIPTION

This work spotlight on clustering problems using medical datasets where there are six well-known public domain
benchmark datasets (see section A) that are presented in UCI machine learning repository (http://archive.ics.uci.edu/ml/index.html) are used to evaluate the performance of the proposed approaches in this work by using two way of calculation the minimal distance (see section B). For further information on medical clustering problems, please refer to [12], [2].

A. Benchmark Datasets

In general, there are varied datasets in terms of the records number and attributes that have different complexity. All of the datasets information about the diseases that are taken from a real infected patients and denoted for research purposes. . The number of clusters in all of the datasets tested here is recommended by the provider of the datasets, except one dataset was recommended [2]. Datasets are summarized as follow:

**Dataset 1:** Wisconsin Breast Cancer Database (B.C) with:
- number of instances: 699
- number of attributes: 10 (including the class attribute)
- attributes type: integers
- number of clusters used in *K-Means* is 3 (i.e. 123, 240 and 363).

**Dataset 2:** Lung Cancer Database (L.C) with:
- number of instances: 32
- number of attributes: 56
- attributes type: integers
- number of clusters used in *K-Means*: 3 (i.e. 9, 13 and 10).

**Dataset 3:** BUPA Liver Disorders Database (B.L.D) with:
- number of instances: 345
- number of attributes: 7
- attributes type: integer, categorical and real
- number of clusters used in *K-Means*: 2 (i.e. 145 and 200).

**Dataset 4:** Pima Indian Diabetes Database (P.I.D) with:
- number of instances: 768
- number of attributes: 8
- attributes type: integer and real
- number of clusters used in *K-Means*: 2 (i.e. 500 and 268).

**Dataset 5:** Haberman’s Survival Database (H.S) with:
- number of instances: 306
- number of attributes: 4
- attributes type: integer
- number of clusters used in *K-Means*: 2 (i.e. 225 and 81).

**Dataset 6:** Thyroid gland data Disease Database (T.D) with:
- number of instances: 215
- number of attributes: 21
- attributes type: categorical and real
- number of clusters used in *K-Means*: 3 (i.e. 150, 30 and 35).

B. Cluster Quality Calculation

In cluster analysis, it is very important issue to evaluate the clustering results quality that is produced by a certain measure. Those measures can be used to compare solutions obtained from different algorithms and also can be used to guide some optimization search processes to find the best partitioning procedure which fits the underlying dataset [14], [15] and [16].

In this work, two functions are used to measure the clusters and their centers i.e.:

i. **Between Objects**

   This method depends on the calculation on the distance (cluster quality) between each data pattern. So we produced a method for clustering in which samples are added in if they are “close” to at least one sample in the candidate cluster.

ii. **Between Centers**

   This method depends on calculating the cluster quality (distance) using the sum of distance between each of data pattern and the cluster center that it belongs to.

III. THE ALGORITHM

In this work, we use two (i.e. N1 and N2) neighborhood structures (see section A) to propose the hybridization between Iterative Simulated Annealing (ISA) and Modified Great Deluge Algorithm (MGD) algorithm is presented for solving clustering problems (ISA-MGD). (see section B).

A. Neighborhood Structures

In this work, two neighborhood structures are used which are adopted from [17] (coded as N1 and N2). The description of the employed neighborhood structures are given as follows:

N1: Randomly selects one pattern from each cluster to swap their data as in [12].

N2: Randomly select two different patterns from the same cluster and swap their data as in [12].

B. Hybridization between iterative simulated annealing and modified great deluge algorithms

In this work, the hybridization between Iterative Simulated Annealing (ISA) and Modified Great Deluge Algorithm (MGD) algorithm is presented for solving clustering problems (ISA-MGD). This hybridization aims to incorporate the strength of one approach with the strength of the other hoping a more promising algorithm. Also this combination can help to diverse the search space.

The Iterative Simulated Annealing (ISA) is proposed in
[12] to overcome the limitation of the original Simulated Annealing (SA) [1]. ISA structure resembles SA structure, but the basic difference is in term of the stopping condition. In ISA, when the temperature (Temp) is less or equal to the final temperature, we reinitialize the temperature (Temp) equal to the initial temperature (T₀). This process of reinitializing the temperature will be repeated in ISA, until no improvement obtained on the best solution (S_{\text{strange}}). Where, Modified Great Deluge (MGD) is proposed in [13] to overcome the limitation of the original GD [13]. MGD the original structure of the GD algorithm, but the basic difference is in term of updating the level. In MGD, we have introduce a list that keeps the previous level value at the time when the better solution is obtained (i.e. S_{\text{strange}} = S_{\text{working}}). When the maximum number of iteration of no improved GD (not_improving_length_GD) is met, then the level is updated by a new level that is randomly selected from the list (where the size of the list is set to 10 based on preliminary experiments).

In this work, a hybridization between iterative simulated annealing and the modified great deluge algorithms are applied, where the algorithm starts with a given K-Means partitions. The notations used in this work are listed as below:

- S₀ : initial solution
- f(S₀) : quality of S₀
- T₀ : initial temperature
- T_f : final temperature
- Temp : current temperature, where at first is set to T₀
- α : decreasing rate
- S_{\text{strange}} : best solution
- f(S_{\text{strange}}) : the quality of S_{\text{strange}}
- S_{\text{source}} : the current solution
- f(S_{\text{source}}) : the quality of S_{\text{source}}
- S_{\text{working}} : the candidate solution
- f(S_{\text{working}}) : the quality of S_{\text{working}}
- level : boundary
- est.q : estimated quality of the final solution
- N.\text{iters} : number of iterations
- Iterations : iteration counter
- β : decreasing value
- LL : list to store level values
- not_improving_length_GD : maximum number of iterations where there is not improvement in the quality of the solution

The same parameters as those employed in ISA [12] and MGD [13] are imposed in this work, where the initial temperature T₀ is equal to 10, and the final temperature T_f is 0. At the beginning of the search, Temp is set to be T₀, at every iteration the temperature Temp is decreased by α, where α is equal to 0.7. The level is set to be an initial water level. The level is decreased by β in each of the iterations, where β is based on the estimated quality (est.q). The pseudo code for the ISA-MGD for clustering problems is shown in Fig. 1.

From Fig. 1, the algorithm starts by initializing the required parameters as in Step-1 by setting the stopping condition (N.\text{iters}), the temperature (Temp) is equal to the initial temperature (T₀), initialize the decreasing temperature rate (α), initialize the estimated quality of the final solution (est.q), the initial water level (level), decreasing rate (β), maximum number of not improving solutions (not_improving_length_GD) and a list of LL size to store the value of the level. Note that, the initial solution is generated using K-Means (S₀).

**Procedure ISA-MGD Algorithm**

**Step 1:** Initialization Phase
- Determine initial candidate solution S₀ and f(S₀);
- S_{\text{strange}} = S₀ ; f(S_{\text{strange}}) = f(S₀);
- S_{\text{source}} = S₀ ; f(S_{\text{source}}) = f(S₀);
- Set N.\text{iters}; (stopping condition)
- Set estimated quality of final solution, est.q;
- Set not improving length GD, not_improving_length_GD;
- Set initial temperature T₀;
- Set initial water level (LL), decreasing rate β = ( (f(S₀) - est.q ) / (N.\text{iters}) );
- Set initial temperature T₀;
- Set final temperature T_f;
- Set decreasing temperature rate as α, where α = 0.7;
- Set list size, to store level values into list LL;
- Iterations = 0; not_improving_counter = 0;

**Step 2:** Improvement (Iterative) Phase

**repeat** (while termination condition is not satisfied)

**Step 2.1:** Selecting candidate solution S_{\text{working}}
- Generate candidate solutions by applying neighbourhood structure (N1 and N2) and the best solution consider as candidate solution (S_{\text{working}});

**Step 2.2:** Accepting Solution
- if f(S_{\text{working}}) < f(S_{\text{strange}});
- S_{\text{strange}} = S_{\text{working}} ; f(S_{\text{strange}}) = f(S_{\text{working}});
- S_{\text{source}} = S_{\text{working}} ; f(S_{\text{source}}) = f(S_{\text{working}});
- Update LL by the level value (FIFO);
- not_improving_counter = 0;
- else
- δ = f(S_{\text{working}}) - quality (S_{\text{source}});
- Generate a random number called RN between 0 and 1;
- if RN ≤ e^{-δ/\text{Temp}};
- S_{\text{source}} = S_{\text{working}};
- else
- if f(S_{\text{working}}) ≤ level;
- S_{\text{source}} = S_{\text{working}};
- else
- Increase not_improving_counter by one;
- if not_improving_counter = not_improving_length_GD;
- Temp = T_f;
- Level = random level from the list LL;
- end if
- if Temp = Temp-Temp*α;
- level = level - β;
- end if
- Iterations = Iterations + 1;
- until Iterations > N.\text{iters} & Temp < T_f (termination conditions are met)

**Step 3:** Termination phase
- Return the best found solution S_{\text{strange}}.

**end Procedure**

Figure 1. Pseudo code for hybridize modified great deluge algorithm with iterative simulated annealing for medical data clustering problems.

In the improvement phase (Step-2), basically the initial solution is iteratively improved by employing the hybridization method (ISA-MGD) until the stopping condition is met. In Step-2.1, neighbourhood structures N1 and N2 are applied to generate candidate solutions (in this case, five candidate solutions are generated), and the best candidate solution is selected as S_{\text{working}}. There are two cases to be taken into account such as:

- Case 1: Better solution
  - If f(S_{\text{working}}) is better than f(S_{\text{strange}}), then S_{\text{working}} is accepted as a current solution (S_{\text{source}} ← S_{\text{working}}), the best
solution is updated (\(S_{\text{source}} \leftarrow S_{\text{working}}\)), and the list value is updates (FIFO) by level (LL \(\leftarrow\) level) as shown in Step-2.2. The Temp will be decreased by the value \(\alpha\) (i.e. Temp = Temp- Temp*\(\alpha\)) and the level will be updated by the value \(\beta\) (i.e. level = level - \(\beta\)).

- **Case 2:** Worse solution
  If \(f(S_{\text{working}})\) is less than \(f(S_{\text{source}})\), then the difference between the quality of \(S_{\text{working}}\) and \(S_{\text{source}}\) is calculated. A random number \([0, 1]\), \(RN\), is generated. If the probability (i.e. \(e^{-0.5}\), where \(\delta = f(S_{\text{working}}) - f(S_{\text{source}})\)) is less than or equal to \(RN\) then \(S_{\text{working}}\) is accepted, and the current solution is updated (\(S_{\text{source}} \leftarrow S_{\text{working}}\)). Otherwise, the quality of \(S_{\text{working}}\) is compared against the level. If it is less than or equal to the level, then \(S_{\text{working}}\) is accepted, and the current solution is updated (\(S_{\text{source}} \leftarrow S_{\text{working}}\)). Otherwise, \(S_{\text{working}}\) will be rejected. The level will be updated by the value \(\beta\) (i.e. level = level - \(\beta\)). The counter for not improving solution is increased by 1. If this counter is equal non_improving_length_GD, then we reinitialize the temperature (Temp) equal to the initial temperature (\(T_0\)), and the level is updated by a new level that is randomly selected from the list (where the size of the list is set to 10 based on preliminary experiments). Otherwise, the process continues until the stopping conditions are met (i.e. Iterations \(> N_{\text{Iters}}\) \&\& Temp \(< T_f\)), and return the best solution found \(S_{\text{source}}\) (Step-2). Note that in this work the est.q is set to 0, and non_improving_length_GD is set to 10 (after some preliminary experiments).

The process will continue until the termination conditions are met (Iterations \(> N_{\text{Iters}}\) \&\& Temp \(< T_f\)), and return the best solution found so far \(S_{\text{source}}\) (Step-3).

**IV. RESULTS AND DISCUSSION**

In this work, we ran our algorithm 20 times across 6 datasets. The algorithms are programmed in Java language and are tested on a PC with an Intel dual core 1800 MHz, 2GB RAM. In the analysis part, the terms used are as follows:
- **N-NI:** Number of not improved iterations.
- **N-I:** Number of improved iterations.
- **Std:** Standard Deviation.

The parameters setting used in these experiments are shown in Table I for ISA-MGD algorithm, where it shows that, the parameter setting for ISA such as, \(T_0, T_f\) and \(\alpha\), in addition to the parameters setting for MGD such as \(\text{est.q}, \text{non_improving_length_GD}\), and \(N_{\text{Iters}}\), in addition to \(LL\) parameter that stores the level values as in [13]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_0)</td>
<td>Initial temperature</td>
<td>0</td>
</tr>
<tr>
<td>(T_f)</td>
<td>Final temperature</td>
<td>10</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>Cooling rate</td>
<td>0.7</td>
</tr>
<tr>
<td>(N_{\text{Iters}})</td>
<td>Number of iterations</td>
<td>100,000</td>
</tr>
<tr>
<td>(\text{est.q})</td>
<td>Estimated quality value</td>
<td>0</td>
</tr>
<tr>
<td>(\text{non_improving_length_GD})</td>
<td>Number of not improved solutions</td>
<td>600</td>
</tr>
<tr>
<td>(LL)</td>
<td>Size of list</td>
<td>10</td>
</tr>
</tbody>
</table>

Table II shows the results comparison between IISA, MGD and ISA-MGD algorithms based on the minimal distance calculation (i.e. between objects). The “Avg” represents the average results out of 20 runs. The best results are presented in bold. Table III shows the results comparison between IISA, MGD and ISA-MGD algorithms based on the minimal distance calculation (i.e. between Centers). The “Avg” represents the average results out of 20 runs. Again, the best results are presented in bold.

The results in Table II show that, ISA-MGD algorithm outperform MGD and IISA algorithms on all datasets except in one dataset for each based on the minimal distance calculation (i.e. between objects). Table II also indicates that, ISA-MGD algorithm using N1 and N2 neighborhood structures better than MGD and IISA algorithms using N1 and N2 based on best minimal distance and average score.

The results in Table III show that, ISA-MGD, MGD and IISA algorithms obtain same best result in some instances based on the minimal distance calculation (i.e. between centers). Table III also indicates that, ISA-MGD algorithm using N1 and N2 neighborhood structures better than MGD and IISA algorithms using N1 and N2 in some instances based on average score.

**TABLE II.** Results between Objects obtained from IISA, MGD and ISA-MGD algorithms using N1 and N2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Initial length by K-Means</th>
<th>20 Runs – Minimal Distance Calculated as between Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IISA</td>
</tr>
<tr>
<td>1. B.C.</td>
<td>6379.694</td>
<td>2338</td>
</tr>
<tr>
<td>2. T.D</td>
<td>3178.714</td>
<td>1228.8</td>
</tr>
<tr>
<td>3. B.L.D</td>
<td>17258.715</td>
<td>5771.59</td>
</tr>
<tr>
<td>4. L.C.</td>
<td>182.577</td>
<td>158.98</td>
</tr>
<tr>
<td>5. H.S</td>
<td>2463.972</td>
<td>987.77</td>
</tr>
<tr>
<td>6. P.L.D</td>
<td>100880.390</td>
<td>24920.39</td>
</tr>
</tbody>
</table>
Table IV shows a further analysis on ISA-MGD using N1 and N2 neighborhood structures between objects over all tested datasets. For example, the best results for L.C dataset is 157.8 that is obtained within 12 minutes 13 seconds under 22387 iterations. Meanwhile, the range for minimum and maximum results is in between 157.8 and 160.39. In most of the cases, the results are obtained between 12 minutes 13 seconds to 19 minutes 35 seconds that are considered acceptable.

For example, Fig. 2 shows a 3D scatter graph for K-Means, MGD and ISA-MGD algorithms using N1 and N2 neighborhood structures for the calculation between objects over H.S dataset. Two clusters are represented by the two colors. Fig. 2-a shows that, the two clusters (colors) are mixed with initial minimal distance obtained by K-Means is 2463.972. Whereas, Fig. 2-c shows that, ISA-MGD obtained better improvement in terms of the minimal distance than MGD algorithm which is equal to 1014.05.

<table>
<thead>
<tr>
<th>Dataset Description</th>
<th>B.C</th>
<th>T.D</th>
<th>B.L.D</th>
<th>L.C</th>
<th>H.S</th>
<th>P.I.D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>2090.61</td>
<td>892.57427</td>
<td>5466.33</td>
<td>157.8</td>
<td>1014.05</td>
<td>22919.27</td>
</tr>
<tr>
<td>Iterations for best</td>
<td>98334</td>
<td>99731</td>
<td>98479</td>
<td>22387</td>
<td>99962</td>
<td>99921</td>
</tr>
<tr>
<td>Time</td>
<td>00:44:28</td>
<td>00:11:02</td>
<td>00:03:05</td>
<td>00:12:13</td>
<td>00:01:19</td>
<td>00:12:13</td>
</tr>
<tr>
<td>N-I</td>
<td>58941</td>
<td>60498</td>
<td>53598</td>
<td>14066</td>
<td>20227</td>
<td>62542</td>
</tr>
<tr>
<td>N-NI</td>
<td>38855</td>
<td>38871</td>
<td>45269</td>
<td>85878</td>
<td>79122</td>
<td>34918</td>
</tr>
<tr>
<td>Average</td>
<td>2124.7632</td>
<td>929.28045</td>
<td>5631.021</td>
<td>159.20720</td>
<td>1043.211</td>
<td>24118.18</td>
</tr>
<tr>
<td>Result Range</td>
<td>2090.61</td>
<td>892.57427</td>
<td>5466.33</td>
<td>157.8</td>
<td>1014.05</td>
<td>22919.27</td>
</tr>
<tr>
<td>Iterations Range</td>
<td>98334</td>
<td>99692</td>
<td>9936</td>
<td>99928</td>
<td>98736</td>
<td>99962</td>
</tr>
<tr>
<td>Time Range</td>
<td>00:44:02</td>
<td>00:10:44</td>
<td>00:02:59</td>
<td>00:12:13</td>
<td>00:01:16</td>
<td>00:08:14</td>
</tr>
<tr>
<td>Std</td>
<td>31.357802</td>
<td>20.24604</td>
<td>103.55513</td>
<td>0.71663</td>
<td>21.87092</td>
<td>437.6589</td>
</tr>
</tbody>
</table>

Figure 2. Scatter graph for K-Means, MGD and ISA-MGD algorithms over H.S dataset for minimal distance calculation between objects using N1 and N2 neighborhood structures.
V. CONCLUSION

This work propose hybridize between modified great deluge algorithm and iterative simulated annealing (ISA-MGD) for medical clustering problem, the idea of the hybridization is to divers the search space, after that the performances of the proposed algorithm is compared based on the minimal distance that is calculated based on (i) between objects, and (ii) between centers. The algorithms has been implemented and tested on six well known real datasets. The algorithms in the comparison are our previous proposed modified great deluge algorithm (coded as MGD) and our previous proposed extend iterative simulated annealing (coded as IISA). Two different neighborhood structures are employed within the proposed approaches i.e. (N1 and N2).

Experimental results show that, ISA-MGD algorithm in the minimal distance as between objects calculation performs better compared to the MGD and IISA algorithms based on minimal distance and the average score. Where, ISA-MGD produce good quality solution compared to MGD and IISA algorithms in the minimal distance calculation as between centers.

Generally, it can be concluded that, the algorithms behave differently due to the different measurements imposed during the search process. However, the limitation in MGD, IISA, ISA-MGD algorithms is the neighborhood structures i.e. N1 and N2 are not really effective as it is based at random, which pose a future work.

REFERENCES


